

## **Informe del Proyecto**

**Título del Proyecto: Analisis y Diseños de Algoritmos II**

**Autores:**

- Cristhian Albarracin Zapata 1968253**
- Miguel Angel Ceballos Yate 2259619**
- Nicolas Gutiérrez Ramírez 2259515**
- Karen Jhulieth Grijalba Ortiz 2259623**

**Universidad: Universidad del Valle**

**Fecha: 26 de diciembre de 2024**

## Informe Detallado del Código Proporcionado

### 1. Descripción General

El proyecto implementa un sistema para determinar la mejor ubicación de nuevas sedes universitarias en una matriz bidimensional, maximizando ganancias poblacionales y empresariales, cumpliendo restricciones predefinidas. Utiliza un backend en Node.js que integra un modelo de optimización desarrollado en MiniZinc.

### Archivos Principales

proyecto.js: Conexión con MiniZinc y lógica principal del modelo de optimización.

utilidades.js: Funciones auxiliares para procesamiento y generación de matrices.

index.js: Backend basado en Express.js que expone la funcionalidad mediante una interfaz web.

### 2. Archivo proyecto.js

Este archivo implementa la conexión con MiniZinc y la lógica para calcular la solución óptima.

#### Función Principal: solucion

Entradas:

poblacion: Matriz de población.

empresas: Matriz de entorno empresarial.

ubicacionesExistentes: Coordenadas de las sedes existentes.

n\_sedes: Número de nuevas sedes a ubicar.

Descripción:

Genera matrices de ganancias aplicando restricciones (mínimo poblacional y empresarial).

Llama a conectorMinizinc para resolver el modelo de optimización.

Calcula ganancias iniciales y finales, formateando la salida.

#### Modelo MiniZinc

Variables de Decisión:

universidades: Matriz binaria que indica si se ubica una universidad en una posición.

Restricciones:

Máximo de nuevas sedes.

Prohibición de ubicaciones adyacentes a las actuales.

Condición de ganancias mínimas en zonas seleccionadas.

Función Objetivo:

Maximizar la suma ponderada de población y empresas para las ubicaciones seleccionadas.

### 3. Archivo utilidades.js

Contiene funciones auxiliares para la manipulación de datos y validaciones.

#### Funciones Destacadas

Generación de Matrices:

generarMatrizCuadrada: Crea una matriz inicializada con ceros.

generarMatriz: Genera una matriz con valores aleatorios entre un rango dado.

Cálculo y Validación:

calcular\_ganancia: Suma los valores de una posición y sus vecinos.

esContiguo: Determina si una posición es adyacente a otra.

generar\_matriz\_ganancias: Aplica restricciones para crear una matriz de ganancias.

Conversión y Formateo:

formatearPosiciones: Convierte coordenadas a formato legible.

convertirAFormato: Transforma una cadena de texto en una matriz bidimensional.

convertirUbicaciones: Convierte texto con coordenadas al formato de lista de pares.

### 4. Archivo index.js

Implementa un servidor web utilizando Express.js para interactuar con el sistema.

#### Funcionalidades

Página Principal:

Proporciona un formulario HTML para ingresar datos.

Procesamiento de Datos:

Endpoint: /process.

Convierte las matrices y ubicaciones ingresadas en el formato adecuado.

Valida la estructura y llama a la función solucion para calcular resultados.

Respuesta:

Muestra las ganancias iniciales y finales, así como las ubicaciones seleccionadas, en formato legible.

#### Validaciones

Las matrices deben ser cuadradas y del mismo tamaño.

Las ubicaciones deben estar en el formato correcto.

El número de sedes debe ser un entero positivo.

### 5. Interacción entre los Componentes

El servidor (index.js) recibe los datos ingresados, los valida y llama a las funciones del archivo proyecto.js.

proyecto.js utiliza las utilidades de utilidades.js para procesar los datos antes de invocar MiniZinc.

MiniZinc calcula la solución óptima y devuelve los resultados al backend, que los muestra al usuario.

## 6. Conclusión

El sistema está diseñado para integrar algoritmos de optimización con una interfaz accesible para los usuarios. La estructura modular facilita la reutilización y escalabilidad del código. Cumple con los requisitos del problema presentado en el enunciado del proyecto.

## Posibles Mejoras

- Incorporar pruebas unitarias para las funciones de utilidades.js.
- Implementar manejo de errores más robusto en la interacción con MiniZinc.
- Mejorar la interfaz gráfica para facilitar el uso.

## Referencias

Documentación de MiniZinc: <https://docs.minizinc.org/en/stable/>

Express.js: <https://expressjs.com>